Enabling mutual SSL authentication with HTTPS protocol enabled

Mutual authentication is a secure two-way SSL authentication where users are authenticated with their certificates. To establish a mutual authentication, the authentication server must be configured with HTTPS protocol enabled. This can be done by following the instructions in the section Managing HTTPS /SSL on server.

User certificates can be either self-signed or signed by a trusted certificate authority (CA). The authentication server supports the user certificates installed on browsers and operating systems as well as on Smart Cards. In the latter case, additional configuration is required to enable certificate reading from the Smart Card devices, which is not covered in this manual.

On this page

To enable mutual SSL authentication

1. Create a self-signed CA certificate if the user certificates will be signed by a self-signed Certificate Authority (see Creating a self-signed CA certificate). Not actual for Smart Cards usage.
2. Create a truststore for the authentication server (see Creating an authentication server truststore).
3. Edit the configuration file *config/authserver.properties* and set the appropriate parameters (see Configuring the authentication server).
4. Create a self-signed user certificate if the organisation will use self-signed certificates for users (see Creating a self-signed user certificate). Not actual for Smart Cards usage.
5. Install the user keystore on a user's machine if the self-signed user certificates are used (see Installing a user keystore on the user's machine). Not actual for Smart Cards usage.
6. If the authentication server is deployed on a cluster, see Deployment on cluster.

Creating a self-signed CA certificate

This section only applies if the user certificates will be signed by a self-signed Certificate Authority. If there is a third-party trusted Certificate Authority that signs user certificates, or if your organization already has the infrastructure to do that, please skip this section.

The CA certificate can be generated with the following OpenSSL commands:

```
1. openssl genrsa -des3 -out ca.key 4096
2. openssl req -new -key ca.key -out ca.csr
3. openssl x509 -req -days 365 -in ca.csr -signkey ca.key -
out ca.crt
```

When executing these commands you will be asked for a key password as well as for other information required for the CA certificate. Please read the instructions carefully and provide all required information. If you want the certificate to stay valid for longer than 365 days, you should specify another value in the *-days 365* argument.

Notes for Windows users

- The OpenSSL binaries for Windows operating system can be downloaded from the following website: http://gnuwin32.sourceforge.net/packages/openssl.htm
- All commands should be run with administrator rights in the directory where the openssl executable resides.
- If the error **Unable to load config info** is received, you should specify the path to the OpenSSL configuration file manually, for example:

```
openssl req -new -key ca.key -out ca.csr -config
/path/to/openssl.cnf
```

- The file *openssl.cnf* is by default included in the OpenSSL bundle.

Creating an authentication server truststore

Authentication Server verifies user certificates against CA certificate, which must be stored in a truststore. To create the truststore, keytool executable should be used. It can be found in the JRE or JDK bin directory, for example:

```
C:\Program Files\Java\jre1.8.0_152\bin\keytool.exe
```

To create the truststore for authentication server

1. Run the keytool command with administrator rights to create a keystore with a CA certificate in it:

   ```
   keytool -keystore truststore.jks -import -alias CA_CERTIFICATE_ALIAS -
   file ca.crt
   ```

   ⚠ **Notes**

   - When executing the keytool command you will be asked for a truststore password. Please read the instructions carefully and provide all required information.
   - If the CA provides more than one certificate to be used to sign the user certificates, all of them should be added with the same keytool command. A unique alias should be specified for each certificate.

2. Copy the generated file **truststore.jks** to the *./config* directory of the authentication server.

Configuring the authentication server

To enable certificate authentication, edit the configuration file *config/authserver.properties* and set the appropriate parameters (see **Authentication by certificate** in Authentication server (advance) configuration parameters).

The authentication server handles revoked certificates by reading the certificate revocation list (CRL) files that can be either stored in the file system or available on the web. Usually, the CRL file is provided by the trusted Certificate Authority that signs user certificates. If the CRL file is stored on the file system, specify the following parameter.

```
authentication.certificate.revocation.list.file=/absolute
/path/to/crl/file
```

Otherwise, if the CRL file is available on the web, specify the file URL.

```
authentication.certificate.revocation.list.url=http://url/to
/crl/file
```

After the Authentication Server configuration is updated, restart the service.

Creating a self-signed user certificate

This section only applies if your organization uses self-signed user certificates. Skip this section if there is a third-party Certificate Authority that signs user certificates or if your organization already has the infrastructure to do that.

For starters, you need to create a certificate and a keystore for each user. This can be done with the following OpenSSL commands.

```
1. openssl genrsa -des3 -out user.key 4096
2. openssl req -new -key user.key -out user.csr
3. openssl x509 -req -days 365 -in user.csr -CA ca.crt -
CAkey ca.key -set_serial 01 -out user.crt
```

Note that a serial number of each certificate must be unique.

Next, add a private key and a certificate to the keystore with the following command.

```
4. openssl pkcs12 -export -clcerts -in user.crt -inkey user.
key -out user.p12
```

Notes

- When executing these commands you will be asked for a key and a keystore password as well as other information required for the user certificate. Please read the instructions carefully and provide all required information.
- If you want the certificate to stay valid for longer than 365 days, you should specify another value in the ***-days 365*** argument.

Notes for Windows users

- OpenSSL binaries for Windows operating system can be downloaded from the folowing website: http://gnuwin32.sourceforge.net/packages/openssl.htm.
- All commands should be run with administrator rights in the directory where openssl executable resides.
- If an error "**Unable to load config info**" occurs, you should specify a path to the OpenSSL configuration file manually, for example:

```
openssl req -new -key user.key -out user.csr -config
/path/to/openssl.cnf
```

- The file ***openssl.cnf*** is by default included in the OpenSSL bundle.

Installing a user keystore on the user's machine

The user keystore should be installed on each user's machine. The method of doing this is OS and browser dependent.

Linux

The user keystore can be installed on Linux operating system with a pk12util command from the NSS tools package:

```
pk12util -d sql:$HOME/.pki/nssdb -i user.p12
```

This will make the certificate available system-wide, enabling authentication in Magicdraw and most of the web browsers.

Windows

You can install the user keystore on Windows operating system by double-clicking the keystore file and following the instructions in the import wizard. This way you will make the certificate available system-wide, enabling authentication in MagicDraw and most of the web browsers.

Firefox

You can also install the keystore on a specific web browser, like Firefox, by importing the user keystore:

1. Click **Options**.
2. Click **Advanced**.
3. Click the **Certificates** tab.
4. Click **View Certificates**.
5. Click the **Your Certificates** tab.
6. Click **Import**.
7. Select the **user.p12** keystore.
8. Provide a keystore password and click **OK**.

Chrome

1. Select **Settings**.
2. Click the **Advanced** settings button to expand it.
3. Select **Manage certificates**.
4. Click the **Import** button.
5. Change the file type to **Personal Information Exchange** (*.pfx;*.p12).
6. Select the **user.p12** keystore and click **Next**.
7. Provide a keystore password and click **Next**.
8. Click **Next** > **Finish**.

Deployment on cluster

If the authentication server is deployed on a cluster, all service instances should use the same truststore and configuration parameters. You need to copy the truststore file and the certificate-related parameters from one server node to others.